AMENDMENTS TO THE CLAIMS

Please amend Claims 1-3, 5-6, 8-11, 13, and 21.

Please cancel, without prejudice, Claims 12 and 14-20.

Please add new Claims 22-25.

---

1. (Currently Amended) A computer implemented method of generating an order of loading data into a programmable device comprising the steps of:

a) automatically identifying a plurality of ~~memory cells~~ configuration bits for programming a programmable device by traversing [[in]] a hierarchical schematic representation of ~~an architecture of a~~ the programmable device ~~for which a programming order is desired~~;

b) automatically determining a plurality of addresses corresponding to said plurality ~~memory cells~~ of configuration bits, said plurality of addresses being in an address space of a memory of the programmable device and operable to store configuration bits for programming the programmable device;

c) automatically determining a plurality of logical names for said plurality of ~~memory cells~~ configuration bits; and

d) based on an order in which said address space is traversed ~~plurality of addresses are to be loaded into~~ when programming said programmable device, automatically storing said plurality of logical names for said plurality of ~~memory cells~~ configuration bits within a data structure within computer readable memory, wherein said data structure describes an order in which to program said programmable device.

2. (Currently Amended) The method of Claim 1 wherein step a) comprises the step of:

a1) identifying ~~said plurality of memory cells~~ configuration bits of said plurality of configuration bits which are at the lowest level in said hierarchical schematic representation ~~hierarchy~~.

3. (Currently Amended) The method of Claim 1 wherein step b) comprises the steps of:

b1) determining a wordline associated with ~~one memory cell~~ a configuration bit of said plurality of configuration bits ~~memory cells~~; and

b2) determining a bitline associated with said ~~one memory cell~~ configuration bit of said plurality of configuration bits ~~memory cells~~.

4. (Original) The method of Claim 1 further comprising the step of:

e) repeating said steps a) through d) for each configuration block of said programmable device.

5. (Currently Amended) The method of Claim 1 wherein said d) comprises:

determining whether there is a configuration bit at an address of said plurality of addresses in a configuration block of said programmable device.

6. (Currently Amended) The method of Claim 5 wherein said d) further comprises:

placing a spacer in said data structure of said plurality of logical names [[in]] responsive to a determination that there was no configuration bit at said address in said configuration block.

7. (Original) The method of Claim 1 wherein said programmable device is a complex programmable logic device (CPLD).

8. (Currently Amended) A computer implemented method of generating an order of loading data into a programmable logic device comprising the steps of:

a) accessing a data structure comprising a plurality of logical names corresponding to a plurality of addresses in [[a]] an address space of a memory operable to store configuration bits for programming ~~hierarchical schematic representation of an architecture of~~ a programmable logic device;

b) accessing a data structure specifying an order in which said plurality of addresses are traversed when loading said configuration bits ~~to be loaded~~ into said programmable logic device;

c) automatically ordering said plurality of logical names from step a) based on the order specified in said data structure in step b) and information in the data structure comprising the plurality of logical names corresponding to the plurality of addresses; and

d) automatically storing said ordered plurality of logical names from step c) in a data structure within computer readable memory, wherein said ordered plurality of logical names describe an order of loading said configuration bits [[data]] into said programmable logic device.

9. (Currently Amended) The method of Claim 8 further comprising ~~the step of: e)~~ storing a placeholder in said data structure of said plurality of logical names from step d) if a given address in the address space of the memory operable to store configuration bits does not require a configuration bit.

10. (Currently Amended) The method of Claim 8 further comprising ~~the step of: e)~~ determining whether there is a configuration bit at ~~one address~~ addresses in the address space based on information in the data structure comprising the plurality of logical names corresponding to the plurality of addresses ~~of said plurality of addresses~~.

11. (Currently Amended) The method of Claim 8 ~~wherein step a) further comprises the steps of: a1)~~

further comprising traversing a hierarchical schematic representation of the programmable device to automatically construct the data structure comprising the plurality of logical names corresponding to the plurality of addresses, said traversing comprising:

automatically identifying the configuration bits for programming ~~a plurality of memory cells in said hierarchical schematic representation of said architecture of~~ said programmable device;

[[a2)]] automatically identifying said plurality of addresses corresponding to said ~~plurality of memory cells~~ configuration bits; and

[[a3)]] automatically determining said plurality of logical names for said ~~plurality of memory cells~~ configuration bits.

12. (Cancelled)

13. (Currently Amended) The method of Claim 11 wherein said identifying the configuration bits [[step a)]] comprises ~~the step of: a)~~ identifying said ~~plurality of memory cells~~ configuration bits which are at the lowest level in said hierarchical schematic representation ~~hierarchy~~.

14-20. (Cancelled)

21. (Currently Amended) The computer implemented method of Claim 1, further comprising:

e) receiving a modification to said hierarchical schematic representation of said ~~architecture of said~~ programmable device; and

f) repeating said a) through d) using said modified hierarchical schematic representation ~~of said architecture~~ of said programmable device to automatically generate a new order in which to program said programmable device.

22. (New) A system comprising a processor coupled to a bus and memory coupled to said bus wherein said memory contains processor instructions for implementing a method of generating an order of loading data into a programmable logic device, said method comprising the steps of:

a) accessing a configuration bit data structure comprising a plurality of logical names corresponding to a plurality of addresses in an address space of a memory operable to store configuration bits for programming a programmable logic device;

b) accessing an address order data structure specifying an order in which said address space is traversed when loading said configuration bits into said programmable logic device; and

c) constructing a bit order data structure within computer readable memory that describes an order of loading said configuration bits into said programmable logic device by repeating c1) - c3) for addresses in the address space based on the address order data structure, wherein said c1) - c3) comprise:

c1) determining whether an address in the address space comprises a configuration bit, based on the configuration bit data structure;

c2) automatically storing a space in said bit order data structure responsive to a determination in said c1) that there is no configuration bit at the address in the address space; and

c3) automatically storing the logical name of the configuration bit in the bit order data structure based on the configuration bit data structure, responsive to a

determination in said c1) that there is a configuration bit at the address in the address space.

23. (New) The system of Claim 22, wherein said method further comprises:

d) traversing a hierarchical schematic representation of the programmable device to automatically construct the configuration bit data structure, said traversing comprising:

d1) automatically identifying the configuration bits;

d2) automatically identifying said plurality of addresses corresponding to the configuration bits; and

d3) automatically determining said plurality of logical names for the configuration bits.

24. (New) The system of Claim 23, wherein said automatically identifying the configuration bits comprises identifying the configuration bits which are at the lowest level in said hierarchical schematic representation.

25. (New) The system of Claim 23, further comprising:

receiving a modification to said hierarchical schematic representation of said programmable device;

repeating said d), wherein said modified hierarchical schematic representation is traversed to automatically construct a new configuration bit data structure; and

repeating said a) through c) using said new configuration bit data structure to automatically generate a new order in which to program said programmable device.